



**SRI RAMACHANDRA**

**INSTITUTE OF HIGHER EDUCATION AND RESEARCH**

(Category - I Deemed to be University) Porur, Chennai

**SRI RAMACHANDRA FACULTY OF ENGINEERING AND TECHNOLOGY**

**CSE 454 – Machine Learning and Cloud services – 2022-2023**

**Project Report**

**AWS Services**

**By**

**Santosh Prasad**

**E0119050**

## What is AWS?

Amazon Web Services (AWS) is the world's most comprehensive and widely used cloud platform, with over 200 fully featured services available from data centres around the world. Millions of customers, including the fastest-growing startups, largest enterprises, and leading government agencies, rely on AWS to reduce costs, improve agility, and accelerate innovation.

## Introduction

AWS offers more services and features within those services than any other cloud provider, ranging from infrastructure technologies like compute, storage, and databases to emerging technologies like machine learning and artificial intelligence, data lakes and analytics, and the Internet of Things. This makes it faster, easier, and less expensive to migrate your existing applications to the cloud and build almost anything you can think of.

AWS also has the most comprehensive functionality within those services. For example, AWS provides the widest range of databases that are purpose-built for various types of applications, allowing you to select the best toolset in terms of cost and performance.

## Largest Community

AWS offers more services and features within those services than any other cloud provider, ranging from infrastructure technologies like compute, storage, and databases to emerging technologies like machine learning and artificial intelligence, data lakes and analytics, and the Internet of Things. This makes it faster, easier, and less expensive to migrate your existing applications to the cloud and build almost anything you can think of.

AWS also has the most comprehensive functionality within those services. For example, AWS provides the widest range of databases that are purpose-built for various types of applications, allowing you to select the best toolset in terms of cost and performance.

## Most Secure

AWS is designed to be the most adaptable and secure cloud computing platform available today. Our core infrastructure is designed to meet the security needs of the military, global banks, and other high-risk organisations. This is supported by a comprehensive set of cloud security tools, which includes 230 security, compliance, and governance services and features. AWS supports 90 security standards and compliance certifications, and all 117 AWS services that store customer data support encryption.

## Fastest pace of innovation

With AWS, you can experiment and innovate more quickly by leveraging cutting-edge technologies. We are constantly increasing our rate of innovation in order to create entirely new technologies that you can use to transform your business. AWS, for example, pioneered the serverless computing space in 2014 with the launch of AWS Lambda, which allows developers to run code without provisioning or managing servers. And AWS created Amazon SageMaker, a fully managed machine learning service that enables ordinary developers and scientists to use machine learning with no prior experience.

## Most proven operational expertise

AWS has unrivalled experience, maturity, reliability, security, and performance on which you can rely for your most critical applications. AWS has been providing cloud services to millions of customers worldwide for over 16 years, serving a wide range of use cases. AWS has the most operational experience of any cloud provider at a larger scale.



## Machine Learning on AWS

- Amazon Augmented AI
- Amazon CodeGuru
- Amazon Comprehend
- Amazon DevOps Guru
- Amazon Elastic Inference
- Amazon Forecast
- Amazon Fraud Detector
- Amazon HealthLake
- Amazon Kendra
- Amazon Lex
- Amazon Lookout for Equipment
- Amazon Lookout for Metrics
- Amazon Lookout for Vision
- Amazon Monitron
- AWS Panorama
- Amazon Personalize
- Amazon Polly
- Amazon Rekognition
- Amazon SageMaker
- Amazon SageMaker Ground Truth
- Amazon Textract
- Amazon Transcribe
- Amazon Translate
- Apache MXNet on AWS
- AWS Deep Learning AMIs
- AWS DeepComposer
- AWS DeepLens
- AWS DeepRacer
- AWS Inferentia
- TensorFlow on AWS

These are some of the machine learning services offered by AWS

Most of these models are rigorously trained and tuned to reproduce same results every time. AWS pre-trained AI Services offer pre-built intelligence for your applications and workflows. AI Services integrate seamlessly with your applications to address common use cases like personalised recommendations, modernising your contact centre, improving safety and security, and increasing customer engagement. Because we use the same deep learning technology that powers Amazon.com and our ML Services, you can expect quality and accuracy from continuously-learning APIs. And, best of all, AWS AI Services do not require prior machine learning experience.

## 1. Know your data on any scenario :

Here we have taken a data set of Zomato Bangalore Restaurant from Kaggle.

COLUMNS	DESCRIPTION
URL	This contains the URL of the restaurant on the Zomato website
ADDRESS	This contains the address of the restaurant in Bangalore
NAME	This contains the name of the restaurant
ONLINE_ORDER	Whether online ordering is available in the restaurant or not
BOOK_TABLE	Table book option available or not
RATE	contains the overall rating of the restaurant out of 5
VOTES	contains the total number of upvotes for the restaurant
PHONE	contains the phone number of the restaurant
LOCATION	contains the neighborhood in which the restaurant is located
REST_TYPE	restaurant type
DISH_LIKED	dishes people liked in the restaurant
CUISINES	food styles, separated by comma
COST	contains the approximate cost of a meal for two people
REVIEWS_LIST	list of tuples containing reviews for the restaurant, each tuple consists of two values, rating and review by the customer
MENU_ITEM	contains list of menus available in the restaurant
LISTED_IN(type)	type of meal
LISTED_IN(city)	contains the neighborhood in which the restaurant is listed

## Code

```
1  # Shape of the dataframe
2  zomato.shape
3
4  # Descriptive Analysis of the DataFrame
5  zomato.describe()
6
7  # Analysis of different objects
8  zomato.info()
9
10 # Checking for null values
11 zomato.isnull().sum()
```

## Output

	rate	votes	cost
count	41237.000000	41237.000000	41237.000000
mean	3.702030	352.772001	369.586259
std	0.440034	884.409230	242.522954
min	1.800000	0.000000	1.000000
25%	3.400000	21.000000	200.000000
50%	3.700000	73.000000	400.000000
75%	4.000000	277.000000	500.000000
max	4.900000	16832.000000	950.000000

2. [AWS](#)

## Recognition

## Introduction:

Amazon Rekognition is a 2016 cloud-based software as a service (SaaS) computer vision platform. It has been purchased and used by a number of US government agencies, including US Immigration and Customs Enforcement (ICE) and the Orlando, Florida police, as well as private entities.

## Capabilities

Rekognition provides a number of computer vision capabilities, which can be divided into two categories: Algorithms that are pre-trained on data collected by Amazon or its partners, and algorithms that a user can train on a custom dataset.

As of July 2019, Rekognition provides the following computer vision capabilities.

### Pre-trained algorithms

- Celebrity recognition in images
- Facial attribute detection in images, including gender, age range, emotions (e.g. happy, calm, disgusted), whether the face has a beard or mustache, whether the face has eyeglasses or sunglasses, whether the eyes are open, whether the mouth is open, whether the person is smiling, and the location of several markers such as the pupils and jaw line.
- People Pathing enables tracking of people through a video. An advertised use-case of this capability is to track sports players for post-game analysis.
- Text detection and classification in images
- Unsafe visual content detection

### Algorithms that a user can train on a custom dataset

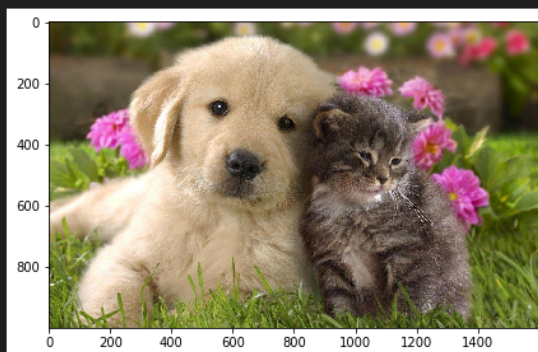
- SearchFaces enables users to import a database of images with pre-labeled faces, to train a machine learning model on this database, and to expose the model as a cloud service with an API. Then, the user can post new images to the API and receive information about the faces in the image. The API can be used to expose a number of capabilities, including identifying faces of known people, comparing faces, and finding similar faces in a database.

## Code:

```
1 import pandas as pd
2 import numpy as np
3 import boto3
4 import skimage.io as io
5 a = pd.read_csv("new_user_credentials.csv")
6 aid = a["Access key ID"][0]
7 akey = a["Secret access key"][0]
8 filename = "catdog.jpg"
9
10 #Initialize the client
11 client = boto3.client('rekognition', aws_access_key_id=aid,
12                                aws_secret_access_key=akey,
13                                region_name='us-west-2')
14 # display the image
15 img = io.imread(filename)
16 io.imshow(img)
17
18 # Detect the objects in the image
19 with open(filename, "rb") as source_image:
20     response = client.detect_labels(Image={'Bytes': source_image.read()})
21     print('Detected labels for catdog.jpg')
22     for label in response['Labels']:
23         print(label['Name'], label['Confidence'])
24
```

## Output:

```
Detected labels for Q1.jpg
Pet 95.64997100830078
Animal 95.64997100830078
Dog 94.59351348876953
Mammal 94.59351348876953
Canine 94.59351348876953
Cat 94.01900482177734
Plant 85.57337951660156
Kitten 84.05640411376953
Grass 83.59941864013672
Golden Retriever 79.39808654785156
Puppy 58.654808044433594
Manx 57.687347412109375
```



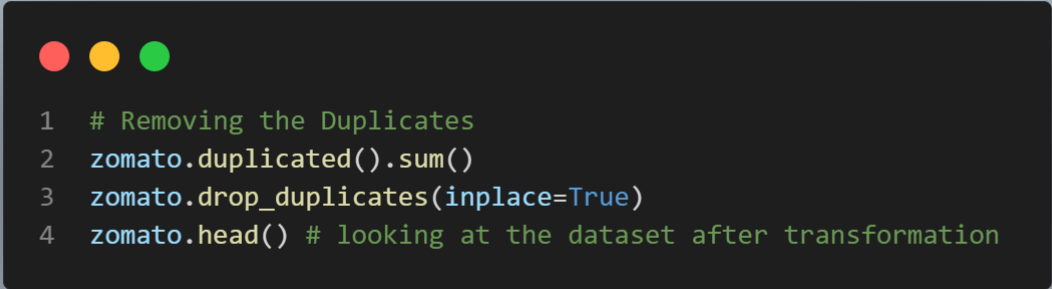


Here we can see that AWS rekognition has correctly classified cat and dog and is displayed in the labels

### 3. Handling Missing Values

- Missing data (or missing values) is defined as the data value that is not stored for a variable in the observation of interest. The problem of missing data is relatively common in almost all research and can have a significant effect on the conclusions that can be drawn from the data
- Missing data present various problems, the absence of data reduces statistical power, which refers to the probability that the test will reject the null hypothesis when it is false
- It can reduce the representativeness of the samples.
- It may complicate the analysis of the study. Each of these distortions may threaten the validity of the trials and can lead to invalid conclusions

#### Removing The Duplicates



```
1 # Removing the Duplicates
2 zomato.duplicated().sum()
3 zomato.drop_duplicates(inplace=True)
4 zomato.head() # looking at the dataset after transformation
```

Output

	address	name	online_order	book_table	rate	votes	location	rest_type	cuisines	a
0	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	
1	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	
2	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	
3	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	South Indian, North Indian	
4	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	

## Dropping Records of NAN values



```

1 # Dropping records of NAN
2 zomato.dropna(how='any',inplace=True)
3 zomato.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 43499 entries, 0 to 51716
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   address                               43499 non-null  object
1   name                                  43499 non-null  object
2   online_order                          43499 non-null  object
3   book_table                            43499 non-null  object
4   rate                                  43499 non-null  object
5   votes                                 43499 non-null  int64
6   location                              43499 non-null  object
7   rest_type                             43499 non-null  object
8   cuisines                              43499 non-null  object
9   approx_cost(for two people)           43499 non-null  object
10  reviews_list                          43499 non-null  object
11  menu_item                             43499 non-null  object
12  listed_in(type)                       43499 non-null  object
13  listed_in(city)                       43499 non-null  object
dtypes: int64(1), object(13)
memory usage: 5.0+ MB

```

## Changing column names

```

1  #Changing the column names
2  zomato = zomato.rename(columns={'approx_cost(for two people)': 'cost', 'listed_in(type)': 'type',
3                                'listed_in(city)': 'city'})
4  zomato.columns

```

```
Index(['address', 'name', 'online_order', 'book_table', 'rate', 'votes',
      'location', 'rest_type', 'cuisines', 'cost', 'reviews_list',
      'menu_item', 'type', 'city'],
      dtype='object')
```

## Column Transformations

```
1 #Some Transformations
2 zomato['cost'] = zomato['cost'].astype(str) #Changing the cost to string
3 zomato['cost'] = zomato['cost'].apply(lambda x: x.replace('/', '.')) #Using lambda function to replace '/' from cost
4 zomato['cost'] = zomato['cost'].astype(float) # Changing the cost to Float
5 zomato.info() # looking at the dataset information after transformation
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43499 entries, 0 to 51716
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
0   address         43499 non-null  object
1   name            43499 non-null  object
2   online_order    43499 non-null  object
3   book_table      43499 non-null  object
4   rate            43499 non-null  object
5   votes           43499 non-null  int64
6   location        43499 non-null  object
7   rest_type       43499 non-null  object
8   cuisines        43499 non-null  object
9   cost            43499 non-null  float64
10  reviews_list    43499 non-null  object
11  menu_item       43499 non-null  object
12  type            43499 non-null  object
13  city            43499 non-null  object
dtypes: float64(1), int64(1), object(12)
memory usage: 5.0+ MB
```

## Removing '/5' from Rates Columns

```

1 #Removing '/5' from Rates
2 zomato = zomato.loc[zomato.rate != 'NEW']
3 zomato = zomato.loc[zomato.rate != '-'].reset_index(drop=True)
4 remove_slash = lambda x: x.replace('/5', '') if type(x) == np.str else x
5 zomato.rate = zomato.rate.apply(remove_slash).str.strip().astype('float')
6 zomato['rate'].head() # looking at the dataset after transformation

```

```

0    4.1
1    4.1
2    3.8
3    3.7
4    3.8
Name: rate, dtype: float64

```

Removing 'yes' or 'no' with 'True' and 'False' in online\_order,book\_table

```

1 # Replacing Yes,No with True,False in online_order,book_table
2 zomato.name = zomato.name.apply(lambda x:x.title())
3 zomato.online_order.replace(('Yes','No'),(True, False),inplace=True)
4 zomato.book_table.replace(('Yes','No'),(True, False),inplace=True)
5 zomato.head()

```

	address	name	online_order	book_table	rate	votes	location	rest_type	cuisines	cost	reviews_list
0	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	True	True	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800.0	[('Rated 4.0', 'RATED\n A beautiful place to ...
1	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	True	False	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800.0	[('Rated 4.0', 'RATED\n Had been here for

## Extracting Reviews from Reviews List column

```
1 # Extracting reviews from the reviews_list column
2 def extract(text):
3     a=''
4     try:
5         a=text.split("\\n")[1].split('"')[0].split("'")[0]
6
7     except IndexError:
8         a=" "
9     return a
10
11 zomato['reviews']=zomato['reviews_list'].apply(extract)
12
13
```

## Encode the Categorical Variables

```
1 #Encode the input Variables
2 def Encode(zomato):
3     for column in zomato.columns[~zomato.columns.isin(['rate', 'cost', 'votes'])]:
4         zomato[column] = zomato[column].factorize()[0]
5     return zomato
6
7 zomato_en = Encode(zomato.copy())
8 zomato_en.head() # looking at the dataset after transformation
```

	address	name	online_order	book_table	rate	votes	location	rest_type	cuisines	cost	reviews_list	menu_item
0	0	0	0	0	4.1	775	0	0	0	800.0	0	0
1	1	1	0	1	4.1	787	0	0	1	800.0	1	0
2	2	2	0	1	3.8	918	0	1	2	800.0	2	0
3	3	3	1	1	3.7	88	0	2	3	300.0	3	0
4	4	4	1	1	3.8	166	1	0	4	600.0	4	0

## 4. Amazon Comprehend

- Amazon Comprehend uses natural language processing (NLP) to extract insights about the content of documents. It develops insights by recognizing the entities, key phrases, language, sentiments, and other common elements in a document.
- For example, using Amazon Comprehend you can search social networking feeds for mentions of products or scan an entire document repository for key phrases.
- All of the Amazon Comprehend features can analyse UTF-8 text documents as the input files. In addition, custom entity recognition can analyse image files, PDF files, and Word files.

Amazon Comprehend uses a pre-trained model to examine and analyse a document or set of documents to gather insights about it. This model is continuously trained on a large body of text so that there is no need for you to provide training data.

Amazon Comprehend gathers the following types of insights:

- **Entities** – References to the names of people, places, items, and locations contained in a document.
- **Key phrases** – Phrases that appear in a document. For example, a document about a basketball game might return the names of the teams, the name of the venue, and the final score.
- **Personally Identifiable Information (PII)** – Personal data that can identify an individual, such as an address, bank account number, or phone number.
- **Language** – The dominant language of a document.
- **Sentiment** – The dominant sentiment of a document, which can be positive, neutral, negative, or mixed.
- **Targeted sentiment** – The sentiments associated with specific entities in a document. The sentiment for each entity occurrence can be positive, negative, neutral or mixed.
- **Syntax** – The parts of speech for each word in the document.

## Data Sample

```
zomato_en["reviews"]
```

```
0      A beautiful place to dine in.The interiors t...
1      Had been here for dinner with family. Turned...
2      Ambience is not that good enough and it
3      Great food and proper Karnataka style full m...
4      Very good restaurant in neighbourhood. Buffe...
...
41232  Ambience- Big and spacious lawn was used to ...
41233  A fine place to chill after office hours, re...
41234  Food and service are incomparably excellent...
41235  Nice and friendly place and staff is awesome...
41236  Great ambience , looking nice good selection...
Name: reviews, Length: 41237, dtype: object
```

## Code

```
1 # User Credentials
2 df=pd.read_csv(r"../Data/new_user_credentials.csv")
3 accessid=df["Access key ID"][0]
4 accesskey=df["Secret access key"][0]
5
6 # Connecting to AWS Comprehend
7 client = boto3.client('comprehend',aws_access_key_id=accessid, aws_secret_access_key=accesskey,region_name='us-east-1')
8
9 # Predicting Sentiment
10 for i,j in enumerate(zomato_en["reviews"]):
11     response = client.detect_sentiment(Text=j,LanguageCode='en')
12     zomato_en["Sentiment"][i] = response["Sentiment"]
```

## Predicted Sentiment

	reviews	Sentiment
0	A beautiful place to dine in.The interiors t...	POSITIVE
1	Had been here for dinner with family. Turned...	POSITIVE
2	Ambience is not that good enough and it	NEGATIVE
3	Great food and proper Karnataka style full m...	POSITIVE
4	Very good restaurant in neighbourhood. Buffe...	POSITIVE



## 5. ML Model

Amazon SageMaker is a fully managed machine learning service. With SageMaker, data scientists and developers can quickly and easily build and train machine learning models, and then directly deploy them into a production-ready hosted environment. It provides an integrated Jupyter authoring notebook instance for easy access to your data sources for exploration and analysis, so you don't have to manage servers. It also provides common machine learning algorithms that are optimized to run efficiently against extremely large data in a distributed environment. With native support for bring-your-own-algorithms and frameworks, SageMaker offers flexible distributed training options that adjust to your specific workflows. Deploy a model into a secure and scalable environment by launching it with a few clicks from SageMaker Studio or the SageMaker console. Training and hosting are billed by minutes of usage, with no minimum fees and no upfront commitments.

### Model Building

#### Code

```
1 # Splitting Independent & Dependent variables
2 X = zomato_en[["online_order", "book_table", "votes", "rest_type", "cuisines", "cost", "type", "city"]]
3 Y = zomato_en["rate"]
4
5 # Train Test Split
6 X_train, X_test, y_train, y_test = train_test_split(X, Y)
7
8 def model(model):
9     # Initializing Linear Regressor
10    mlr=model
11
12    # Train the model
13    mlr.fit(X_train, y_train)
14
15    # Predict the result
16    y_pred = mlr.predict(X_test)
17
18    # Metrics
19    print(f"For {str(model).split('(')[0]}")
20    print("MSE -", mean_squared_error(y_test, y_pred))
21    print("MAE -", mean_absolute_error(y_test, y_pred))
22    print("R^2 -", r2_score(y_test, y_pred))
23    print("\n")
```

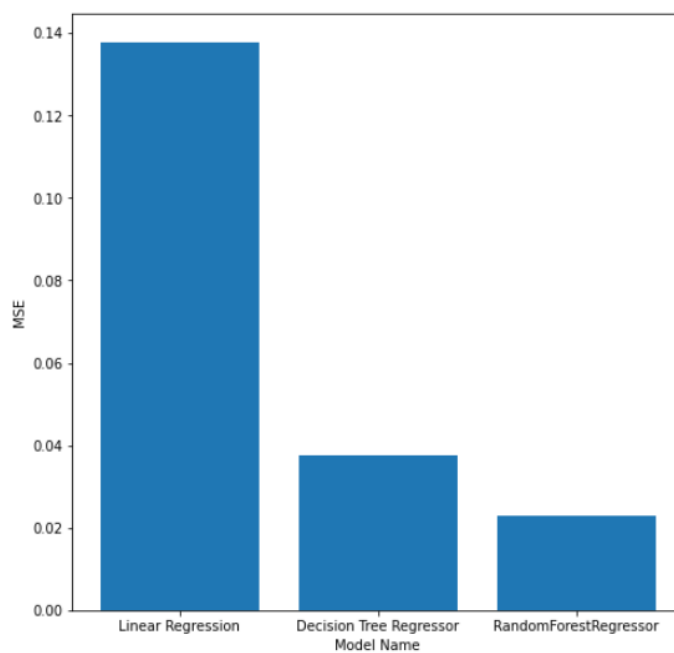
## Output

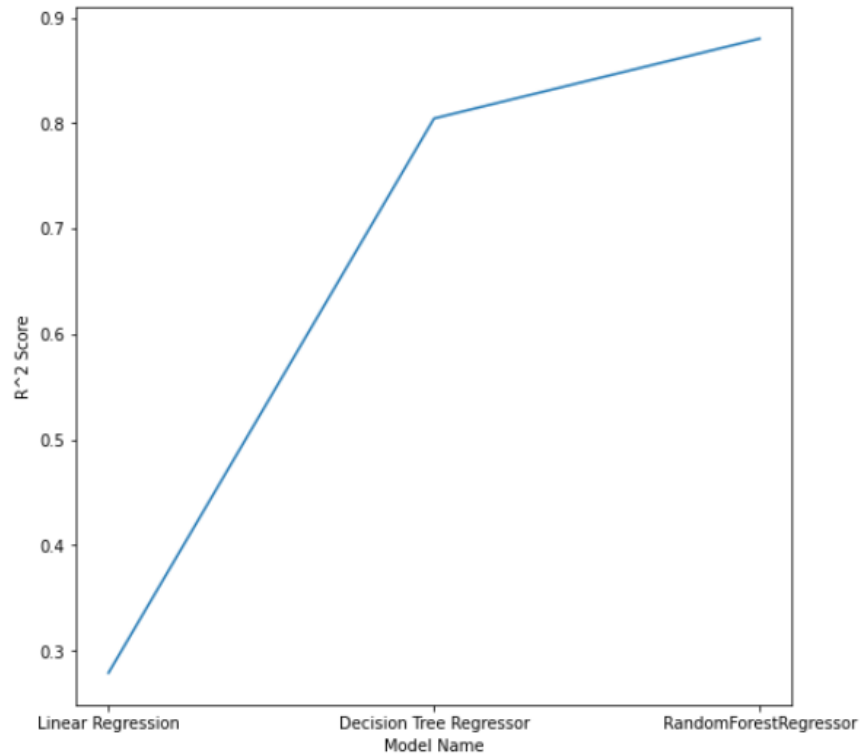
```
For LinearRegression  
MSE - 0.13783860871687673  
MAE - 0.29050081395319516  
R^2 - 0.27900753838649706
```

```
For DecisionTreeRegressor  
MSE - 0.037404946653734235  
MAE - 0.06512124151309417  
R^2 - 0.8043459316990653
```

```
For RandomForestRegressor  
MSE - 0.022949748852512032  
MAE - 0.08223324280633718  
R^2 - 0.8799567401861161
```

## Result and Discussion





## Insights

- The data is modelled on three different Regressor models and compared them based on the performance metrics
- Random Forest model tops the list as it has low MSE value & high coefficient of determination value.
- Since the results on test data is high and does vary a lot with respect to the train accuracy, we can say that the model can make reliable predictions with low bias and variance.

Course completion certificate:

